

**AMENDMENTS TO THE SPECIFICATION:**

Please replace the paragraph beginning at page 5, line 29, with the following rewritten paragraph:

A1  
--Prior art for recording the persistent execution state of a virtual machine for the Java platform can be found in ~~Orthogonal persistence for the Java platform. Draft specification, October 1999, <http://www.sun.com/research/forest/index.html>, M. Jordan and M. Atkinson;~~ and Persistent execution state of a Java Virtual Machine, *Proc. ACM 2000 Java Grande Conference*, San Francisco, June 2000, T. Suezawa. These systems provide support for checkpointing the state of a Java application and virtual machine. They do not store the executable code for various procedures.--

Please replace the paragraph beginning at page 5, line 29, with the following rewritten paragraph:

A2  
--Figure 4 shows a block diagram of a QSI writer (400). In the preferred embodiment of the method, the QSI writer is obtained by modifying a run-time compiler from prior art. In another embodiment, it is obtained by modifying a static, offline compiler. A front-end (401) processes the program to produce an intermediate code representation (402), which is fed to an optimizer (403) that produces optimized intermediate code (404). The front-end and optimizer represent well-known components of prior art compilers, and may be organized in different ways, including, being organized in the form of multiple modules. The method of this invention adds to the optimizer a component (405) to record dependencies between different modules. This component is described further in Figure 5. The optimized intermediate code annotated with dependence information (406) produced by this component

A<sup>2</sup>

is processed by the back-end code generator (407), which may ignore the annotations on dependence information in the process of producing executable binary code (408). The method of this invention adds to the code generator an *adaptation annotation recorder* component (409), described further in Figure 6, which produces a further annotated executable code (410) with annotations to help adapt the code to a new execution context. The *QSI recorder* (411), described further in Figure 7, produces the QSI (412) which is stored for later processing.--